

DIAGNÓSTICO - ESTRUCTURAS DE DATOS - SOLUCIÓN

1. Dado el siguiente código en lenguaje C, señale la afirmación correcta:

```
#include <stdio.h>

int main(void)
{
    float num, *res;

    printf ("Introduzca un numero:\n");
    scanf ("%f", &num);
    res = 2*num;
    printf ("%f", *res);

    return 0;
}
```

- a. **El programa tiene un error que se detecta en tiempo de compilación (no compila)** Verdadero. La variable res es un puntero a una variable de tipo float. Al ser un puntero sólo puede almacenar direcciones de memoria. Al intentar asignar a res el valor 2*num (float) se produce un error de compilación (tipos incompatibles).
- b. **La variable res almacena el doble del valor de num**
- c. **El programa muestra en pantalla el valor acumulado en la variable res**
- d. **El programa muestra por pantalla la dirección de la variable res**

2. Dada la siguiente lista de elementos del lenguaje C: **define, main, +, ***, podemos decir que:

- a. **Todos son operadores.** Falso: define y main son palabras clave o palabras reservadas de C. Por su parte "+" y "*" son, respectivamente, el operador suma y multiplicación
- b. **Los dos primeros son palabras clave y tienen un significado especial para el compilador .** Verdadero
- c. **Solo los dos últimos pertenecen al léxico de C.** Falso. El léxico de C se corresponde con los elementos básicos con los que se construyen los programas. Entre estos elementos se incluyen las palabras clave y los operadores. Por tanto, los cuatro términos que componen la lista pertenecen al léxico de C.
- d. **Los dos primeros son identificadores.** Falso. Los identificadores se corresponden con el nombre dado a las variables y funciones definidas por el programador.

3. Un puntero...

- a. ... es una variable cuyo valor es una dirección de memoria. Correcto
- b. ... es una variable que almacena cualquier valor menor que la memoria (en MB) del disco duro. Falso. Las direcciones almacenadas en un puntero son direcciones de la Memoria Principal.
- c. ... no es una variable pero almacena cualquier valor menor que la memoria (en MB) del disco duro. Falso. Los punteros son variables.
- d. ... no es una variable ya que almacena un valor que es una dirección de memoria. Falso

4. Indique cuál de las siguientes opciones no se corresponde con un tipo de datos en C:

- a. **char**. Verdadero. Es el tipo de datos usado para almacenar caracteres alfanuméricos.
- b. **float** Verdadero. Es el tipo de datos usado para almacenar números reales en coma flotante con precisión simple.
- c. **Null**. Falso. NULL es una macro definida en los archivos de cabecera `stddef.h`, `stdio.h`, `stdlib.h` y `string.h` que se usa para inicializar un puntero cuando queremos que "no apunte a ningún sitio".
- d. **void**. Verdadero. Es un tipo "especial" de datos que se utiliza para indicar que una función no tiene argumentos, que no devuelve ningún valor o para crear punteros genéricos.

5. Indique cuál de los siguientes tipos de datos es estructurado:

Solución: Los tipos de datos estructurados tienen una estructura interna con la que es posible incluir, bajo un mismo tipo, varios elementos. En este grupo se incluyen: las cadenas de caracteres, los vectores y matrices y las estructuras o registros. Por tanto la opción correcta es la a

- a. Cadenas de caracteres.
- b. Enteros (numéricos).
- c. Reales (numéricos).
- d. Caracteres.

6. Indique el resultado del siguiente programa en C:

```
#include <stdio.h>

int main(void) {
    int a = 5, b = 3;
    float c = 0;

    c = a/b;
    printf(" %d entre %d es: %f", a, b, c);
    return 0;
}
```

Solución: Dado que a y b son variables enteras el resultado de su división también será un valor entero: en este caso 1. Al asignar a c (float) un valor entero (1) se

produce una conversión de tipos. En este caso, 1 se transforma a real añadiendo una parte decimal en la que todos los elementos son cero. Por tanto el resultado del programa será el indicado en la opción a.

- a. 5 entre 3 es: 1.000000
- b. 5 entre 3 es: 1
- c. 5 entre 3 es: 1.666666
- d. Se produce un error por el tipo de datos de la variable c.

7. Indique el resultado del siguiente programa en C:

```
#include <stdio.h>

int main(void) {
    int a = 0;
    int *p_a;

    a = 5;
    p_a = &a;
    a = a+20;
    printf("%d", *p_a);
    return 0;
}
```

Solución: "p_a" es un puntero a una variable de tipo entero. La instrucción "p_a=&a" asigna a "p_a" la dirección de memoria en la que está guardada la variable "a". A partir de este momento, "*p_a" es equivalente a "a". Tras ejecutar la sentencia "a = a+20", "a", y por tanto "*p_a" toman el valor 25. Por tanto, la respuesta correcta es la b.

- a. 5
- b. 25
- c. 0
- d. Se produce un error de compilación.

8. Indique qué realiza el siguiente programa en C:

```
#include <stdio.h>

int main(void) {
    int a = 0;
    int *p_a;

    p_a = &a;
    printf("%p", p_a);
    return 0;
}
```

Solución: Al igual que en el ejercicio anterior, "p_a" es un puntero a una variable de tipo entero. La instrucción "p_a=&a" asigna a "p_a" la dirección de memoria en la que está guardada la variable "a". Sin embargo, en este caso, lo que se muestra por pantalla es el valor de la variable puntero, es decir, la dirección de memoria en

la que se encuentra almacenada la variable "a" Por tanto, la respuesta correcta es la b.

- a. Muestra el valor de la variable *a* (es decir, 0).
- b. Muestra la dirección en la que está almacenada la variable *a*.
- c. Se produce un error de compilación porque no se pueden imprimir direcciones del disco duro por pantalla.
- d. Se produce un error de compilación porque la asignación de la dirección de la variable *a* es errónea.

9. Indique cual de las siguientes afirmaciones es verdadera:

- a. Una variable de tipo "int" puede almacenar valores negativos. Verdadero. Las variables de tipo entero almacenan valores enteros tanto positivos como negativos.
- b. Una variable de tipo "float" no puede almacenar valores enteros. Falso. Una variable de tipo float puede almacenar valores enteros (valores cuya parte decimal es 0)-
- c. La sentencia `char letra=b` asigna a la variable `letra` el valor `b`. Falso. Los caracteres siempre se escriben encerrados entre comillas simples. Por tanto, la sentencia correcta sería: `char letra='b'`.
- d. La sentencia `float f=3,2` asigna a la variable `f` el valor `3,2`. Falso. En C, como en la mayoría de los lenguajes de programación, el separador decimal es el punto (.) y no la coma (,). Por tanto, la sentencia correcta sería `float f=3.2`.

10. Dado el siguiente fragmento de código, indique cuál de las siguientes afirmaciones es correcta.

```
#include <stdio.h>
int main(void) {

    int variable=5;
    int resultado;
    resultado= (variable >= 5);
    printf("%d \n", resultado);
    return (0);
}
```

Solución: El resultado de evaluar una expresión relacional es verdadero (1) o falso (0). En este caso puesto que `variable=5`, la relación `variable >= 5` es verdadera y, por tanto, `resultado` toma el valor 1. Por tanto, la respuesta correcta es la a.

- a. En pantalla aparece 1
- b. En pantalla aparece 0
- c. En pantalla no aparece nada
- d. La compilación detecta un error de sintaxis.